Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-12 (2016) 651–655

Linear time algorithm for Precedence Constrained Asymmetric Generalized Traveling Salesman Problem

Alexander Chentsov * **Michael Khachay** * **Daniel Khachay** *

* *Krasovsky Institute of Mathematics and Mechanics, 16 S.Kovalevskoy str., Ural Federal University, 19 Mira str.; Ekaterinburg 620990, Russia (e-mail: agchentsov@mail.com, mkhachay@imm.uran.ru, daniil.khachay@gmail.com)*

Abstract: We consider the combinatorial optimization problem of visiting clusters of a fixed number of nodes (cities), where, on the set of clusters should be visited according to some kind of partial order defined by additional precedence constraints. This problem is a kind of the Asymmetric Generalized Traveling Salesman Problem (AGTSP). To find an optimal solution of the problem, we propose a dynamic programming based on algorithm extending the well known Held and Karp technique. In terms of special type of precedence constraints, we describe subclasses of the problem, with polynomial (or even linear) in n upper bounds of time complexity.

Keywords: Asymmetric Generalized Traveling Salesman Problem (AGTSP), NP-hard problem, dynamic programming

1. INTRODUCTION AND RELATED WORK

The generalized traveling salesman problem (GTSP) extends the well-known traveling salesman problem (TSP), where the set of cities is partitioned into disjoint clusters, and the salesman has to visit every cluster exactly once. The problem has numerous applications, e.g. in carrier-vehicle routing (Garone et al., 2014) and Nuclear Power Plant dismantling (Chentsov and Chentsov, 2001).

There are multiple approaches to finding of optimal and suboptimal solutions of this problem. First approach is to reduce the considered instance of GTSP to some appropriate instance of regular Traveling Salesman Problem (TSP). According to (Laporte et al., 1987), there is a cost-preserving reduction of GTSP to asymmetric TSP, i.e. for the initial problem, the researchers can use the diversity of algorithms and solvers developed for the classic TSP (Helsgaun, 2015; Karapetyan and Gutin, 2011). Unfortunately, the resulting TSP has a very general structure, it is not even a metric one, so, to approximate this problem we could not use efficient algorithms like famous Christofides $3/2$-approximation algorithm (Christofides, 1975), Arora's PTAS (Arora, 1998) for the Euclidean TSP or even PTAS for Euclidean multiple salesman problems (Khachay and Neznakhina, 2015; Khachai and Neznakhina, 2015).

Another approach is of adopting some kind of evolutionary techniques: genetic algorithms (Bontoux et al., 2010; Gutin and Karapetyan, 2010), ant colony (Jun-man and Yi, 2012), etc. According to published results of numerical evaluations, in some cases, this approach yields good approximate solutions efficiently. But the main shortcoming of this approach is lack of theoretical support, since all these heuristics have no approximation guarantees and theoretical upper bounds of time complexity.

On the other hand, for the classic TSP, there are many well-described polynomial time solvable special cases (see e.g. (Deineko et al., 2014)). Investigating the similar cases of GTSP seems to be also very perspective. However, to the best of our knowledge, there are no publications presenting results in this field. In this paper, we try to bridge such a gap. Basically, our results can be considered as a simple extension of the results obtained in (Balas, 1999) for the classic TSP.

We consider the most general case of the GTSP. In this setting, for any pair of incident nodes u and v, traveling costs for the forward and backward transitions are not supposed to be the same. To emphasize this *asymmetry*, we call this setting the Asymmetric Generalised Traveling Salesman Problem (or AGTSP).

Precedence constraints appear to be a regular component of the AGTSP instances induced by real-life applications. These constraints define an order for the clusters to visit and can be easily supplemented by a natural interpretation in terms of object domain.

For example, in the problem of high-precision laser cutting of a metal sheet, it is required to cut off metal pieces of a complicated shape. In corresponding AGTSP instance, each shape is represented by a finite cluster of pierce-points where cutting process can be suspended or resumed. As it is shown at Fig. 1, the shapes can be embedded to each other, so, the most inner objects should be cut first. This order induces natural precedence constraints on a given set of clusters (Fig. 2).

The rest of the paper is organized as follows. In Section 2, we provide a mathematical statement of the considered Asymmetric Generalized Traveling Salesman Problem. Further, in Section 3, we recall the famous Held-Karp dynamic programming procedure used for finding the exact solution of the problem in question. The main point here is that traveling and city visiting

10.1016/j.ifacol.2016.07.767

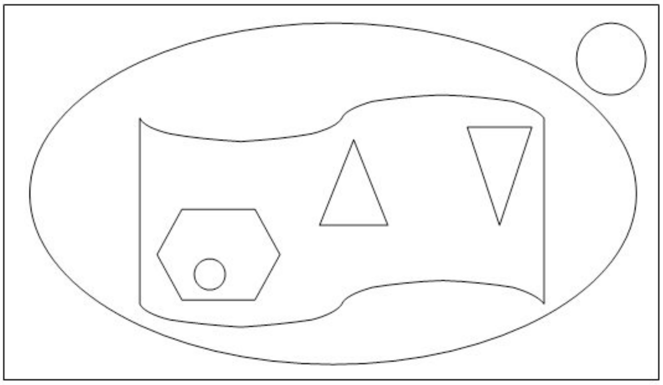

Fig. 1. Shape cutting problem

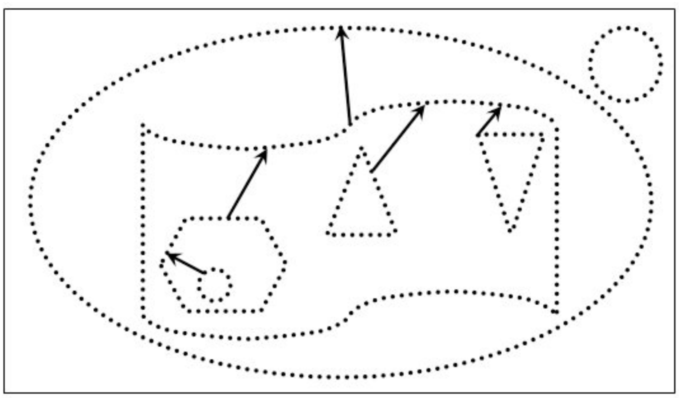

Fig. 2. Order of cutting produces precedence constraints

costs in our case depend on partial subtours and DP procedure can successfully overcome this issue. In Section 4 we show that this procedure can be easily reformulated in terms of finding the cheapest s-t-paths in corresponding weighted acyclic digraph (also known as digraph of states). Our subsequent results presented in Section 5 are based on this representation of the dynamic programming procedure. Finally, in Section 6 we present simple example confirming the applicability of the precedence constraints considered.

2. PROBLEM STATEMENT

We consider the extended setting of the Asymmetric Generalized Traveling Salesman Problem (AGTSP) (Fig. 3). Input: finite disjunctive sets (clusters) $M_1,\dots,M_n$ of nodes to be visited and a dedicated start point $x_0 \notin \cup M_i$. Without loss of generality, we assume that all clusters have the same number $p \geq 1$ of nodes:

$$M_j = \{g_{j1},\dots,g_{jp}\}.$$

Transition costs $\hat{c}(x_0, g_{j\tau})$ and $\check{c}(g_{j\tau}, x_0)$ for moves from the point x_0 to any $g_{j\tau}$ (and vice versa) are given along with costs $c(g_{l\sigma}, g_{j\tau})$ for any

$$j,l \in \mathbb{N}_n = \{1,\dots,n\},\ j \neq l \text{ and } \sigma,\tau \in \mathbb{N}_p.$$

For any cluster, the visiting cost $c'(g_{j\tau})$ (which can be interpreted as expenses of inner job) is given as well. As usual, the problem is to find the cheapest tour starting and finishing in the point x_0 and visiting every cluster once. Actually, it is required to find a permutation

$$\pi : \mathbb{N}_n \to \mathbb{N}_n$$

defining the visiting order for the clusters and the finite sequence

$$g_{\pi(1)\tau(1)},\dots,g_{\pi(n)\tau(n)}$$

(also known as g-tour), such that

$$\hat{c}(x_0, g_{\pi(1)\tau(1)}) + \sum_{i=1}^{n-1} \left(c'(g_{\pi(i)\tau(i)}) + c(g_{\pi(i)\tau(i)}, g_{\pi(i+1)\tau(i+1)})\right) + \check{c}(g_{\pi(n)\tau(n)}, x_0) \to \min \quad (1)$$

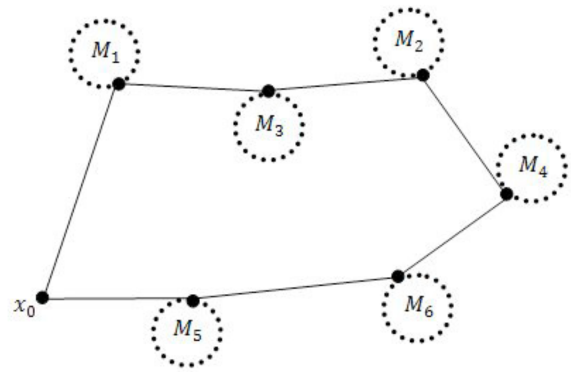

Fig. 3. An instance of the AGTSP for $n = 6$.

The main differences between the problem studied in this paper and the standard setting of the AGTSP are as follows:

(i) for any nodes $g_{l\sigma}$ and $g_{j\tau}$, the transition cost $c(g_{l\sigma}, g_{j\tau})$ and the cluster visiting cost $c'(g_{j\tau})$ depend on the chosen sub-tour connecting x_0 and the node $g_{l\sigma}$;

(ii) on the set of clusters, there is defined one of two types of additional Balas precedence constraints (like proposed in (Balas, 1999) for the regular TSP):

Type I. For a natural number $k \leq n$, any feasible permutation π satisfies the equation

$$\forall i,j \in \mathbb{N}_n\ \ (j \geq i+k) \Rightarrow (\pi(i) < \pi(j)). \quad (2)$$

Type II. For any natural values

$$1 \leq k(1),\dots,k(n) \leq n$$

and any feasible permutation π,

$$\forall i,j \in \mathbb{N}_n\ (j \geq i+k(i)) \Rightarrow (\pi(i) < \pi(j)). \quad (3)$$

Actually, it can be seen that constraint (2) can be obtained from (3), where $k(i) = k$ for some fixed value k.

3. DYNAMIC PROGRAMMING

We start with the description of the proposed dynamic programming method, which goes back to fundamental results by Bellman (Bellman, 1962) and Held & Karp (Held and Karp, 1961). Suppose, the optimal g-tour sourcing from x_0 and visiting for the first $i-1$ turns the clusters with indexes from $J \subset \mathbb{N}_n$, in the i-th turn, visits the cluster M_j at the node $g_{j\tau(i)} \in M_j$. Denote the cost of this i-turns g-subtour by

$$C(J,i,j,g_{j\tau(i)}).$$

Then, the following recursive equations hold

$$C(\varnothing,1,j,g_{j\tau(1)}) = \hat{c}(x_0, g_{j\tau(1)}), \quad (4)$$

$$C(J,i,j,g_{j\tau(i)}) = \min_{l\in J} \min_{g_{l\tau(i-1)}\in M_l} \{C(J\setminus\{l\}, i-1, l, g_{l\tau(i-1)}) + c(g_{l\tau(i-1)}, g_{j\tau(i)}) + c'(g_{j\tau(i)})\}. \quad (5)$$

Further, the optimum of the given instance (1) of AGTSP can be found by the formula

$$C^* = \min_{j\in\mathbb{N}_n} (C(\mathbb{N}_n\setminus\{j\}, n, j, g_{j\tau(n)}) + \check{c}(g_{j\tau(n)}, x_0)). \quad (6)$$

Finally, an optimal g-tour can be easily obtained by backward search procedure.

4. GRAPH REPRESENTATION

The recursive procedure (4)-(6) can be represented equivalently in terms of graph theory. Indeed, assign to the instance of problem (1) the corresponding instance of the cheapest *s*-*t*-path problem in the appropriate $(n+2)$-layered edge-weighted digraph

$$G^*[p] = (V^*[p], A^*[p], w^*[p]),$$

whose vertexes are states of the dynamic programming scheme. Denote by $V_i^*[p]$ the vertex-set of the *i*-th layer, which is defined by

$$V_0^*[p] = \{s\}, V_{n+1}^*[p] = \{t\},$$

where

$$V_i^*[p] = \{(J,i,j,\tau) : j\in\mathbb{N}_n\setminus J,\ g_{j\tau}\in M_j,\ J\subset\mathbb{N}_n, |J| = i-1\} \quad (i\in\mathbb{N}_n). \quad (7)$$

The vertexes s and t are assigned to the starting point x_0; any vertex (state) (J,i,j,τ) corresponds to *i*-turns subtour of the g-tour visiting clusters with indexes $J\cup\{j\}$, wherein the latter visited cluster is M_j (at the node $g_{j\tau}$). In the graph $G^*[p]$, only vertexes of subsequent layers $V_i^*[p]$ and $V_{i+1}^*[p]$ can be adjacent. Moreover, s is adjacent to any vertex from $V_1^*[p]$; any vertex from $V_n^*[p]$ is adjacent to t. Any other states

$$(J,i,l,\sigma) \text{ and } (J', i+1, j, \tau)$$

are adjacent if

$$|J| = i-1,\ J' = J\cup\{l\},\ j\notin J',\ \sigma,\tau\in\mathbb{N}_p. \quad (8)$$

We denote the set of arcs connecting $V_i^*[p]$ with $V_{i+1}^*[p]$ by $A_{i,i+1}^*[p]$. Their weights are defined by the following equations

$$w^*[p](s,(\varnothing,1,j,\tau)) = \hat{c}(x_0, g_{j\tau}),$$

$$w^*[p]((\mathbb{N}_n\setminus\{j\}, n, j, \tau), t) = \check{c}(g_{j\tau}, x_0),$$

$$w^*[p]((J,i,l,\sigma),(J',i+1,j,\tau)) = c(g_{l\sigma}, g_{j\tau}) + c'(g_{j\tau}).$$

It is easy to show that the set of feasible g-tours in (1) is isomorphic to the set of *s*-*t*-paths in the graph $G^*[p]$. Moreover, any corresponding g-tour and *s*-*t*-path have the same costs (weights). Therefore, the cheapest g-tour can be found in $O(|A^*|)$ by the well known modification of the Ford-Bellman algorithm for circuit-free weighted digraph (see, e.g. (Cormen et al., 2009)).

Unfortunately, for the general case of AGTSP, the number of arcs in the graph $G^*[p]$ is growing exponentially as $n\to\infty$, which implies exponential time complexity of the proposed scheme of dynamic programming. Indeed, for any $n\geq 2$

$$|V^*[p]| > |V_2^*[p]\cup\ldots\cup V_n^*[p]| \geq pn2^{n-2}.$$

Moreover, an indegree of any vertex

$$(J,m,j,\tau)\in V_m^*[p] \text{ for } m\geq 2$$

satisfies the equation

$$\deg^-(J,m,j,\tau) = (m-1)p \geq p.$$

Hence,

$$|A^*[p]| = \Omega(np^2 2^n).$$

Nevertheless, taking into account the additional constraints on the set of clusters, e.g. of precedence type (Steiner, 1990), we can drastically decrease the overall time complexity of our optimization procedure. In the following Section 5, we discuss the precedence constraints of Type I and Type II, for which the scheme (4)-(6) has linear (in n) time complexity (for any fixed k and p).

5. COMPLEXITY BOUNDS

We proceed with description of the graphs $G^*[p]$ corresponding to two special cases of AGTSP precedence constraints of Type I and Type II mentioned above.

First, we show that structure of the $G^*[p]$ in general case is completely defined by the structure of the graph $G^*[1]$.

Lemma 1. For any $p > 1$,

$$V_i^*[p] = V_i^*[1]\times\mathbb{N}_p \quad (i\in\mathbb{N}_n) \quad (9)$$

$$A_{0,1}^*[p] = A_{0,1}^*[1]\times\mathbb{N}_p,\ A_{n,n+1}^*[p] = A_{n,n+1}^*[1]\times\mathbb{N}_p \quad (10)$$

$$A_{i,i+1}^*[p] = A_{i,i+1}^*[1]\times\mathbb{N}_p^2 \quad (i\in\mathbb{N}_{n-1}) \quad (11)$$

Indeed, given by an arbitrary $p > 1$ define the mapping

$$\Gamma : V^*[p]\to V^*[1]$$

by the equations

$$\Gamma(s) = s,\ \Gamma(t) = t,\ \Gamma((J,i,j,\tau)) = (J,i,j).$$

Since, for any p, incidence between vertexes of the graph $G^*[p]$ is defined by equation (8), the mapping Γ is a homomorphism. Moreover, vertices

$$(J,i,l,\sigma) \text{ and } (J\cup\{l\}, i+1, j, \tau)$$

are incident in the graph $G^*[p]$ if and only if the vertices

$$(J,i,l) \text{ and } (J\cup\{l\}, i+1, j)$$

are incident in $G^*[1]$ as well.

By construction,

$$\Gamma^{-1}((J,i,j)) = \{(J,i,j,1),\ldots,(J,i,j,p)\},$$

from which validity of equations (9)–(11) follows.

Corollary 2. For any $p > 1$,

$$|A^*[p]| \leq |A^*[1]|p^2.$$

In (Balas, 1999), the structure of graphs $G^*[1]$ defining dynamic procedure for the regular TSP with additional precedence constraints (2) and (3) was described. We summarize these results in Theorem 3.

Theorem 3.

1. In the case of precedence constraints (2),

$$|A^*[1]| = O(n\cdot k^2 2^{k-2}).$$

2. In the case of constraints (3)

$$|A^*[1]| = O(\sum_{i=1}^n k^*(i)(k^*(i)+1)2^{k^*(i)-2})$$

for

$$k^*(i) = \max\{k(j) : i-k(j)+1\leq j\leq i\}.$$

Our main complexity results follow from Lemma 1 and Theorem 3.

Theorem 4. Let for the instance of AGTSP precedence constraint (2) be valid. Then, dynamic programming scheme (4)-(6) obtains an optimal solution (for this instance) in time of

$$O(n \cdot p^2 k^2 2^{k-2}). \tag{12}$$

Theorem 5. If any instance of AGTSP satisfies precedence constraint (3), then running time of scheme (4)-(6) is

$$O(p^2 \sum_{i=1}^{n} k^*(i)(k^*(i)+1)2^{k^*(i)-2}). \tag{13}$$

Theorem 4 and Theorem 5 claim that, in the case of additional precedence constraints (2) or (3), AGTSP can be solved to optimality efficiently. Indeed, it is seen that upper bound (12) (the case of bound (13) can be considered by analogy) is linear in n for any fixed k and p and remains polynomial for $p = O(\mathrm{poly}(n))$ and $k = O(\log(n))$. Therefore, dynamic programming procedure in both cases can find an optimal solution in time depending linearly on number of clusters n.

6. INDUSTRIAL EXAMPLE

We would like to discuss the applicability of use the precedence constraints. At glimpse, constraints (2)-(3) seem to be excessively restrictive. Nevertheless, even the more strict constraint (2) covers all of complexity cases of AGTSP as k varies from 1 to n. Indeed, if $k = 1$, the only feasible permutation is identical. On the other hand, when k tends to n, almost all permutations are feasible.

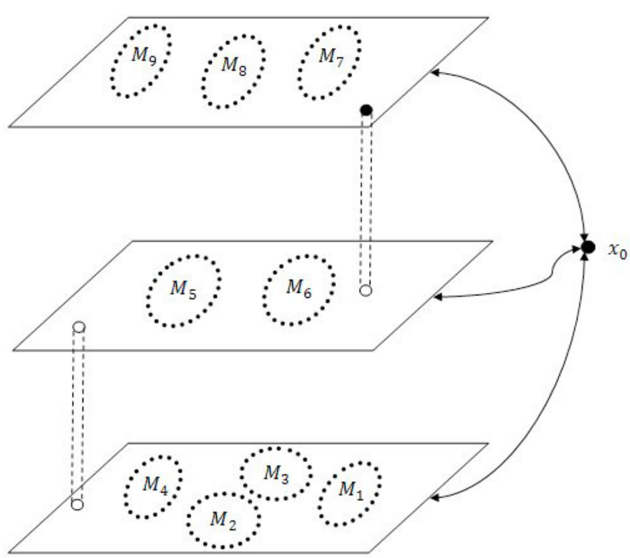

Fig. 4. Illustration of fire rescue mission plan for $m = 3$, $q_1 = 4$, $q_2 = 2$, and $q_3 = 3$

To illustrate the methodology proposed, consider the following industrial application. This application is conserned with planning of a fire rescue mission for a skyscaper building. The skyscaper consists of m floors. Each t-th floor is a set of q_t apartments M_i having several doors to enter (see Fig. 4). The rescue squad can start its mission from any floor, to which it can be delivered for the vanishing cost. When all survivors are found and secured, the squad can be evacuated also from any floor. The main restriction is that moving from one floor to another can be done only through dedicated elevators and any transition costs much more, than any moves around the floor. Formulating such a model mathematically, we obtain the following precedence constraints. Actually, for the apartment M_i, we define

$$k(i) = \begin{cases} q_1 + 1 - i, \text{ if } 1 \le i \le q_1, \\ q_1 + q_2 + 1 - i, \text{ if } q_1 + 1 \le i \le q_1 + q_2, \\ \dots \\ \sum_{l=1}^{m-1} q_l + 1 - i, \text{ if } \sum_{l=1}^{m-2} q_l + 1 \le i \le \sum_{l=1}^{m-1} q_l. \end{cases}$$

Basically, these constraints mean that building should be rescued either bottom up or vice versa. Thus, for this application, precedence constraints like (3) appear to be quite natural.

CONCLUSION

We propose dynamic programming procedure for finding an optimal solution for AGTSP. For two types of precedence constraints, we show that this procedure is efficient. Actually, its time complexity is linear in n for any fixed k and p, and remains polynomial for $k = O(\log n)$ and $p = O(\mathrm{poly}(n))$.

ACKNOWLEDGEMENTS

This research was supported by the Russian Science Foundation, grant No. 14-11-00109.

REFERENCES

Arora, S. (1998). Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45.

Balas, E. (1999). New classes of efficiently solvable generalized traveling salesman problems. *Annals of Operations Research*, 86(0), 529–558.

Bellman, R. (1962). Dynamic programming treatment of the travelling salesman problem. *J. ACM*, 9(1), 61–63.

Bontoux, B., Artigues, C., and Feillet, D. (2010). A memetic algorithm with a large neighborhood crossover operator for the generalized traveling salesman problem. *Computers & Operations Research*, 37(11), 1844 – 1852.

Chentsov, A. and Chentsov, A. (2001). Dynamic programming method in the generalized traveling salesman problem: The influence of inexact calculations. *Mathematical and Computer Modelling*, 33(8–9), 801 – 819.

Christofides, N. (1975). Worst-case analysis of a new heuristic for the traveling salesman problem. In *Symposium on New Directions and Recent Results in Algorithms and Complexity*, 441.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C. (2009). *Introduction to Algorithms*. MIT, 3 edition.

Deineko, V.G., Klinz, B., Tiskin, A., and Woeginger, G.J. (2014). Four-point conditions for the TSP: The complete complexity classification. *Discrete Optimization*, 14, 147–159. doi:10.1016/j.disopt.2014.09.003.

Garone, E., Determe, J.F., and Naldi, R. (2014). Generalized traveling salesman problem for carrier-vehicle systems. *Journal of Guidance Control and Dynamics*, 37, 766–774.

Gutin, G. and Karapetyan, D. (2010). A memetic algorithm for the generalized traveling salesman problem. 9(1), 47–60.

Held, M. and Karp, R.M. (1961). A dynamic programming approach to sequencing problems. In *Proceedings of the 1961 16th ACM National Meeting*, ACM '61, 71.201–71.204. ACM, New York, NY, USA.

Helsgaun, K. (2015). Solving the equality generalized traveling salesman problem using the Lin–Kernighan–Helsgaun algorithm. *Mathematical Programming Computation*, 1–19.

Jun-man, K. and Yi, Z. (2012). Application of an improved ant colony optimization on generalized traveling salesman problem. *Energy Procedia*, 17, Part A(0), 319 – 325.

Karapetyan, D. and Gutin, G. (2011). Lin–Kernighan heuristic adaptations for the generalized traveling salesman problem. *European Journal of Operational Research*, 208(3), 221 – 232.

Khachai, M. and Neznakhina, E. (2015). A polynomial-time approximation scheme for the euclidean problem on a cycle cover of a graph. *Proceedings of the Steklov Institute of Mathematics*, 289(1), 111–125. doi: 10.1134/S0081543815050107.

Khachay, M. and Neznakhina, K. (2015). Approximability of the minimum-weight k-size cycle cover problem. *Journal of Global Optimization*. doi:10.1007/s10898-015-0391-3. URL `http://dx.doi.org/10.1007/s10898-015-0391-3`.

Laporte, G., Mercure, H., and Nobert, Y. (1987). Generalized travelling salesman problem through n sets of nodes: the asymmetrical case. *Discrete Applied Mathematics*, 18(2), 185 – 197.

Steiner, G. (1990). On the complexity of dynamic programming for sequencing problems with precedence constraints. *Annals of Operations Research*, 26(1), 103–123.